Citation

Communications of the ACM >archive
Volume 40 , Issue 12 (December 1997) >toc

Managing multimedia information in database systems

Author
William I. Grosky

Publisher
ACM Press New York, NY, USA

Pages: 72 - 80 Periodical-Issue-Article
Year of Publication: 1997
ISSN:0001-0782

doi> http://doi.acm.org/10.1145/265563.265574 (Use this link to Bookmark this page)

FULL TEXT: **Access Rules**

pdf 1.91 MB

REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 Surajit Chaudhuri , Luis Gravano, Optimizing queries over multimedia repositories, ACM SIGMOD Record, v.25 n.2, p.91-102, June 1996

2 C. Faloutsos , R. Barber , M. Flickner , J. Hafner , W. Niblack , D. Petkovic , W. Equitz, Efficient and effective querying by image content, Journal of Intelligent Information Systems, v.3 n.3-4, p.231-262, July 1994

3 Gemmell, D., Vin, H., Kandlur, D., Rangan, P., and Rowe, L. Multimedia storage servers: A tutorial. IEEE Comp~t. 28, 5 (May 1995), 40-49.

4 Grosky, W. Multimedia information systems. IEEE M~ltimedia 1, 1 (Spring 1994), 12-24.

5 Grosky, W., Fotouhi, F., and Jiang, Z. Using metadata for the intelligent browsing of structured media objects. In

Managing M~ltimedia Data: Using Metadata to Integrate and Apply Digital Media, A. Sheth and W. Klas, Eds. McGraw-Hill, New York, 1997, pp. 67-92.

6 H. V. Jagadish, Content-based indexing and retrieval, The handbook of multimedia information management, Prentice-Hall, Inc., Upper Saddle River, NJ, 1997

7 Ramesh Jain, NSF workshop on Visual Information Management Systems, ACM SIGMOD Record, v.22 n.3, p.57-75, Sept. 1993

8 Ramamritham, K., and Chrysanthis, P. Executive Briefing: Advances in Conc~rrency Control and Transaction Processing. IEEE Computer Society, Los Alamitos, Calif., 1997.

9 Stonebraker, M. Object-Relational DBMSs: The Next Great Wave. Morgan- Kaufmann, San Francisco, 1996.

10 Thimm, H., and Klas, W. ~-sets for optimized reactive adaptive playout management in distributed multimedia database systems. In Proceedings of the 12th IEEE International Cona&rence on Data Engineering (New Orleans, Feb. 26-Mar. 1, 1996). IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 584-592.

11 Thimm, H., and Klas, W. Playout management in multimedia database systems. In Design and Implementation of M~ltimedia Database Management Systems, K. Nwosu, P. Berra, and B. Thuraisingham, Eds., Kluwer Academic Publishers, Boston, 1996, 318-376.

12 Bhavani Thuraisingham, Multilevel security for information retrieval systems—II, Information and Management, v.28 n.1, p.49-61, Jan. 1995

CITINGS

Ronald Fagin, Fuzzy queries in multimedia database systems, Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, p.1-10, June 01-04, 1998, Seattle, Washington, United States

INDEX TERMS

Primary Classification:
H. Information Systems
- H.2 DATABASE MANAGEMENT
  - H.2.4 Systems
    - Subjects: Query processing

Additional Classification:
H. Information Systems
- H.2 DATABASE MANAGEMENT
  - H.2.4 Systems
    - Subjects: Transaction processing
- H.3 INFORMATION STORAGE AND RETRIEVAL
- H.5 INFORMATION INTERFACES AND PRESENTATION (I.7)

General Terms:
Design, Languages

Peer to Peer - Readers of this Article have also read:

Editorial pointers
Communications of the ACM 44, 9
Diane Crawford

News track
Communications of the ACM 44, 9
Robert Fox

Forum
Communications of the ACM 44, 9
Diane Crawford

Foundations of multimedia database systems
Journal of the ACM (JACM) 43, 3
Sherry Marcus , V. S. Subrahmanian

A model of multimedia information retrieval
Journal of the ACM (JACM) 48, 5
Carlo Meghini , Fabrizio Sebastiani , Umberto Straccia

The large data size, structure, and time dependencies of multimedia calls for new processing beyond the abilities of traditional database architectures.

MANAGING MULTIMEDIA INFORMATION IN DATABASE SYSTEMS

William I. Grosky

In the past decade, the database field has been quite active, discovering more efficient methods for managing alphanumeric data; bringing application-dependent concepts, such as rules, into database environments; and managing such new types of data as images and video [4]. When new types of data are first brought into a database environment, it is quite natural that the data needs to be transformed so it is representable in existing database architectures. Thus, when images were first managed in a database, researchers developed numerous techniques for representing them, first in a relational architecture, then in an object-oriented architecture.

In the relational architecture, where an image and its contents were represented as sets of tuples over several relations, researchers initially believed that most of the classic relational techniques developed for indexing, query optimization, buffer management, concurrency control, security, and recovery would work well in the intended environments of the various systems. It was only after some experience working with these new types of data that this approach was shown to have an inherent weakness: a mismatch between the nature of the data and the way both the user and the system were forced to query and operate on it.

Object SQL queries and operations just won't do for multimedia data, for which browsing is an important paradigm. And standard indexing approaches do not work for content-based queries of multimedia data. Other modules of database systems likewise have to be changed in order to manage multimedia data efficiently. Today we realize that this evolution of standard database modules has to be done, but we are far from agreeing on how to do it. Commercial object-relational database systems [9] are the state of the art for implementing multimedia database systems, but even these systems leave much to be desired in such areas as playout management and intuitive querying environments.

Over the past 15 years, managing multimedia data in a database environment has evolved through the following sequence of conceptual and performance insights:

- Multimedia data was first transformed into relations in an ad-hoc way. Only certain types of queries and operations were efficiently supported. Initially, a query, such as "Find all images contain-

Figure 1. Semcons, database objects, and multimedia objects

ing the person shown dancing in this video," was extremely difficult, if not impossible, to respond to efficiently.

- When the weaknesses of this approach became apparent, researchers asked what types of information should be extracted from images and videos and how this information should be represented to support content-based queries most efficiently. The result was a large body of work on multimedia data models.
- Since these data models specified the types of information that could be extracted from multimedia data, the nature of multimedia queries was also discussed. Earlier work on feature matching from the field of image interpretation was brought to bear, helping launch the field of multimedia indexing. Multimedia indexing, in turn, started the ball rolling toward multimedia query optimization techniques.
- A multimedia query was seen as quite different from a standard database query and closer to queries in an information-retrieval setting. The implications of this important concept have still not played themselves out.

These steps made possible investigation into improving other database system modules, research fields that are still in their infancy.

This article covers multimedia data management from the point of view of database systems, focusing on how the various aspects of database design and the modules of database systems have evolved over the years to better manage multimedia data, as well as what the future seems to hold. It proposes several technological advances that must occur for commercial databases to efficiently manage multimedia information in a production environment.

The Nature of Multimedia Data

Multimedia data, consisting of alphanumeric, graphics, image, animation, video, and audio objects, is quite different from standard alphanumeric data in terms of both presentation and semantics. From a presentation viewpoint, multimedia data is huge and involves time-dependent characteristics that must be adhered to for coherent viewing. Whether a multimedia object already exists or is constructed on the fly, its presentation and the user's subsequent interaction with it push the boundaries of traditional database systems.

Because of its complex structure, multimedia data requires complex processing to derive semantics from its contents. Real-world objects shown in images, video, animations, or graphics and discussed in audio participate in meaningful events whose nature is often

the subject of queries. Using state-of-the-art techniques from the fields of image interpretation and speech recognition, systems can often be made to recognize similar real-world objects and events by extracting (with a human in the loop) certain information from the corresponding multimedia objects. This information consists of objects called "features," which are usually less complex and voluminous than the multimedia objects themselves.

How the logical and physical representation of multimedia objects are defined and relate to each other, as well as what features are extracted from these objects and how extraction is accomplished, is in the domain of multimedia data modeling.

Multimedia Data Modeling

In standard database systems, a data model is a collection of abstract concepts that can be used to represent real-world objects, their properties, their relationships to each other, and the operations defined over them. These abstract concepts are capable of being physically implemented in the given database system. Through the mediation of this data model, queries and other operations over real-world objects are transformed into operations over abstract representations of these objects, which are, in turn, transformed into operations over the physical implementations of the abstract representations. What makes a multimedia data model different from a traditional data model is that multimedia objects are completely defined in the database and contain references to other real-world objects that should also be represented by the data model. For example, the person Bill is a real-world "object" that should be represented in a data model. The video "Bill's Vacation" is a multimedia object whose structure as a temporal sequence of image frames should also be represented in the same multimedia data model. However, when Bill is implemented in a database by a given sequence of bits, the sequence is not actually Bill, who is a person. On the other hand, the sequence of bits implementing the video "Bill's Vacation" in the database can be considered to be the actual video. In addition, the fact that Bill appears in various frames of the video "Bill's Vacation" performing certain actions should also be represented in the same data model.

Various types of information should be captured in a multimedia data model, including:

- The detailed structure of the various multimedia objects
- Structure-dependent operations on multimedia objects
- Properties of multimedia objects
- Relationships between multimedia objects and real-world objects
- Portions of multimedia objects with representation relationships with real-world objects, the representation relationships themselves, and the methods used to determine them
- Properties, relationships, and operations on real-world objects

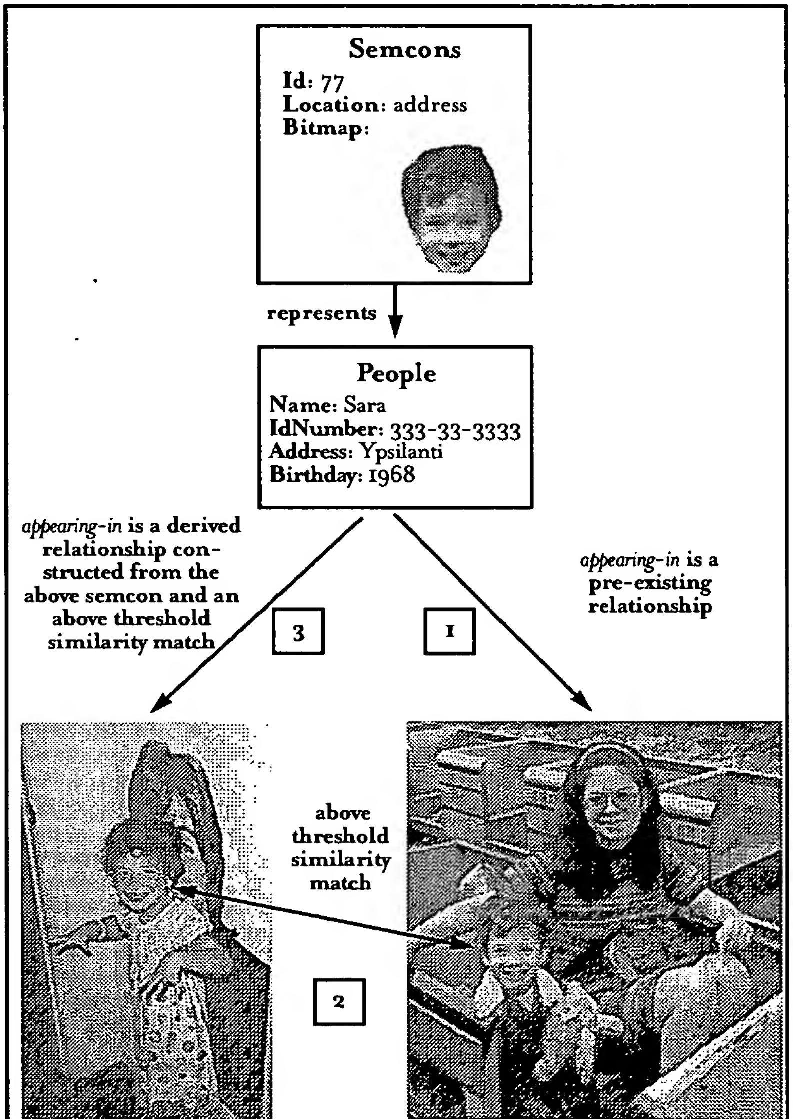

Figure 2. Consequences of a similarity match

For images, the structure should include such things as the image format, the image resolution, the number of bits per pixel, and any compression information; for a video object, such items as duration, frame resolution, number of bits per pixel, color model, and compression information are included. Modeling the structure of a multimedia object is important for many reasons, not least that operations that are structure dependent are defined on these objects. These operations are used to create derived multimedia objects (such as image edge maps) for similarity matching, as well as various composite multimedia objects (such as multimedia presentations) from individual component multimedia objects.

An example of a multimedia object property is the name of the object; for example, "Bill's Vacation" is the name of a particular video object. A relationship between a multimedia object and a real-world object would be the *stars-in* relationship between the actor Bill and the video "Bill's Vacation."

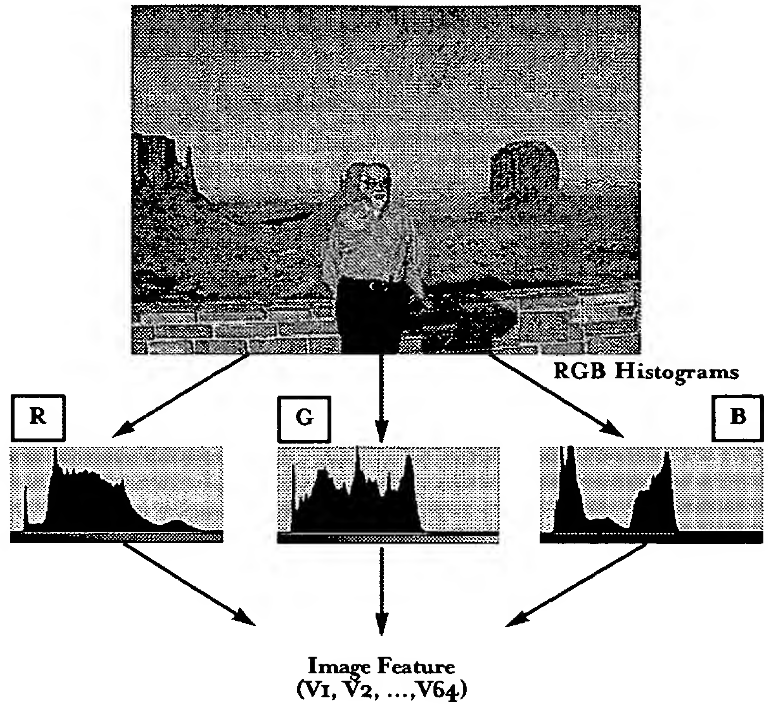

Figure 3. Color histogram of an image represented as a high-dimensional point

Suppose the Golden Gate Bridge is a real-world object being represented in the database and that a particular region of frame six of "Bill's Vacation" is known to show this object. This small portion of the byte span of the entire video is considered to be a first-class database object, called a *semcon* [5], for *icon*ic data with *sem*antics. Therefore, both the *represents* relationship between this semcon and the Golden Gate Bridge object and the *appearing-in* relationship between the Golden Gate Bridge object and "Bill's Vacation" should be captured by the data model. Attributes of this semcon are the various features extracted from it that can be used for similarity matching over other multimedia objects. Semcons can be time-independent, as in the Golden Gate Bridge-Bill's Vacation example, or time-dependent, in which case they correspond to events. Figure 1 includes several image semcons.

There is currently a dearth of tools for multimedia data modeling. If multimedia information is to be intelligently and efficiently managed, this situation has to change. Without question, the continued development of the MPEG-7[1] standard on the content-based description of multimedia data will spur development of such tools. The MPEG-4 encoding methodology is already hierarchical in nature, providing a rudimentary structural decomposition of multimedia objects in which our description of semcons could be represented. MPEG-7, on the other hand, will describe the various types of descriptors that can be associated with these semcons.

[1]http://www.cselt.stet.it/mpeg

Multimedia and Database Systems

The architecture of a standard database system consists of modules for query processing, transaction management, buffer management, file management, recovery, and security. Implementations differ depending on whether the database system is relational/object-oriented or centralized/distributed, but the natures of these modules are basically the same.

Query processing. Querying in a multimedia database is quite different from querying in standard alphanumeric databases. Besides the fact that browsing takes on added importance in a multimedia environment, queries can contain multimedia objects input by the user; the results of these queries are based not on perfect matches but on degrees of similarity.

In a multimedia repository connected to a database system, a user typically initiates exploratory browsing interspersed with various queries. These queries are

typically of the sort that ask for the description of the real-world object *o* corresponding to a semcon *s* initiated by clicking the mouse over *s*, as well as by navigating to other multimedia objects containing semcons similar to *s* or whose represented real-world objects are in some relationship with *o* [5].

Queries entailing retrieval of multimedia objects with a certain property, such as depiction of a desert scene, or inclusion of a representation of a real-world entity also represented in a different multimedia object, cannot be implemented efficiently in a standard database system. Examples of similarity queries are:

1. Retrieve all video shots showing my friend Tom dancing, given a photograph of Tom.
2. Show me all mug shots of criminals resembling this sketch.

The results of such queries are based on similarity matches, not exact matches. What is actually being searched for is multimedia objects corresponding to the same real-world object (see Figure 2 for a derived *appearing-in* relationship constructed from a preexisting *appearing-in* relationship followed by an above-threshold similarity match). It is extremely rare that two images of the same entity match in an exact manner. Similarity measures between two multimedia objects are usually real-valued, ranging from 0 (completely different) to 1 (exactly the same). Theoretically, the result of query 1 should be all video shots in the entire database, each one ranked from 0 to 1 for its similarity to a shot of Tom dancing; the result of query 2 should be all images in the entire database, each one ranked from 0 to 1 for its similarity to the given sketch. Typically, however, there is a specified threshold, so that if the ranking of a given multimedia object is lower than the threshold value, it is not retrieved. Implementation of these operations usually consists of the use of a specialized index via a filtering operation to remove below-threshold multimedia objects from further consideration followed by an ordering based on the rank of the multimedia objects that are left.

Indexes of standard database systems are designed for the standard data types of integers, decimal numbers, floating-point numbers, and character strings, as well as for some date and time data types. They are one-dimensional and usually are hash-based or utilize some of the B-tree variants. In most cases, they are unsuitable for similarity matching.

Over the years, many specialized indexes have been designed for various types of features that cannot be used in traditional database systems [6]. Only database systems that support extensible data types and their associated access methods can be profitably used for these applications. At present, such database systems are called object-relational [9]. Viewed as object-oriented software systems, the associated methods used for multimedia retrieval are quite ad hoc. As MPEG-7 matures, organization of these methods, commonly collected into a group called a "blade" or a "cartridge," will also mature. A standard methodology will emerge combining elementary methods into more complex combinations, depending on the search criteria.

A generic indexing technique is to extract *n* numerical-valued features from a multimedia object and represent these *n* values by an *n*-dimensional point. A spatial index that supports nearest-neighbor searching is then used for similarity matching. These *n* features may be independent of each other or derived from a composite global feature. An example of a composite-feature technique is representation of the color histogram of an image as a high-dimensional point used in the QBIC database system [2] (see Figure 3). While this generic technique is universal, various specialized, more efficient indexing methodologies may develop for particular types of features. An interesting line of research will be the automatic combination of indexing methodologies for individual elementary features to make a single index for a complex feature based on these individual features.

Query optimization is the process of choosing the optimal access path to answer a query. Object-relational database systems supporting nearest-neighbor and user-defined access methods need to know the associated costs involved in using these methods to make the appropriate decision about how to proceed. The translations of various user-defined functions in terms of lower-level access methods, such as those related to nearest-neighbor searching, must also be made known to the system [1]. As the set of elementary methods, along with various ways of combining them, become standardized, these cost functions will be known automatically. An example function would be *desert_scene*, which takes as an argument an image and returns *true* if the similarity of the image to a desert scene is above some fixed threshold. Such a function may be part of an SQL query resulting from various user input actions.

Transaction management. Users interact with database systems through the mediation of transactions. Standard transactions satisfy the four ACID properties:

- *Atomicity*. A transaction is executed atomically. That is, either all of the transaction executes or none of it executes. The former occurs when the transaction commits, the latter if the transaction aborts.

- *Consistency.* Assuming a transaction starts executing when the database is in a consistent state, after it commits, it leaves the database in a consistent state.
- *Isolation.* Each transaction executes in isolation from other transactions. That is, a transaction cannot read the intermediate results of other transactions.
- *Durability.* The results of a committed transaction are made permanent in the database, irrespective of any database failures.

For advanced applications, such as multimedia databases, conventional concurrency control algorithms can be used; the results would still be correct. However, the concurrency of the overall system would suffer, since in this environment, transactions tend to be long, compute-intensive, interactive, and cooperative and to refer to many database objects [8]. Referring to many database objects is illustrated through a transaction referring to a video. If the entire video is locked for an update transaction that inserts subtitles, then many thousands of image frames are also locked, decreasing the throughput for other transactions referring to the same video. Multimedia data is very large, making it impractical to create multiple copies of the data, as is necessary in the versioning approach to concurrency control. Optimistic methods of concurrency control during transactions involving multimedia presentations are also not suitable, as the possible abortion of such a transaction would present difficulties to the user/viewer.

In order to increase system concurrency in such an environment, new transaction models defined for object-oriented environments; long, cooperating activities; and real-time database applications could be used. Current multimedia databases rely on an object-oriented data model, and transactions in such a database can be long and cooperating, such as the computer-supported cooperative work (CSCW) authoring environment, as well as exhibit some real-time factors, as in multimedia presentations.

In order to increase concurrency in these environments, the traditional ACID properties have been generalized. *Atomicity* is changed to *recovery*, which refers to placing the database in a correct state in the event of a database failure or transaction abortion [8]. This recovery can be applied in a nested transaction environment in which a transaction consists of subtransactions, each of which can commit before the entire parent transaction commits. Thus, if the parent transaction aborts due to the abortion of one or more subtransactions, some subtransactions may still have affected the state of the resulting database. Such behavior can take place easily in a CSCW authoring environment.

Consistency need not depend on the traditional concept of serializability; a nonserializable schedule can still leave the database in a consistent state. An example of being left in a consistent state is when two transactions both write the same values into a variable. Another approach is realizing that even though a nonserializable schedule may leave the database in an inconsistent state, the inconsistent state may not be fatal in the long run. If a few contiguous frames of a video presentation have been changed in an imperceptible way by another transaction, such subtle changes usually would not cause a problem.

Isolation is changed to *visibility.* Transactions are allowed to view the results of other transactions. In a cooperative CSCW multimedia-authoring environment, it is important for one person to see what others are doing.

And finally, *durability* is changed to *permanence.* Intermediate results may be written in temporary files that can be shared among users in a cooperative CSCW multimedia-authoring environment that would be destroyed after the complete session is over.

Almost every multimedia database researcher and practitioner agrees that in current multimedia database environments, updates are rare. In a traditional database environment, managing read-only transactions is trivial. However, multimedia data presents another interpretation, called *playout management*, of the concept of read-only transaction management [11]. Presenting a composite multimedia object for user viewing is quite complicated in a multiuser client/server environment, even with local caching. Composite multimedia objects comprise distributed component multimedia objects having various spatiotemporal constraints and typically take a long time and sophisticated buffer management schemes at the server(s) to deliver high Quality of Service (QoS). An added difficulty is that different multimedia objects can share the same component objects. Presentation of a composite multimedia object can thus be considered a transaction. Similar to schedulers in a standard database environment, a scheduler here has to define the execution history of the individual steps making up the construction of a given composite multimedia object.

Thus, it is possible that one presentation blocks another presentation, just as one conventional transaction can block another transaction accessing some of the same database values. User interaction with the presentation further complicates the process. An approach called *reactive adaptive playout management* [10] was recently developed to optimize the playout behavior in multiuser client/server environments by

accounting for how much performance degradation each user can tolerate.

Transaction management for multimedia data will mature only when there is general agreement as to the type of operations to be supported on multimedia data. Knowledge of the semantics of these operations will enable us to more effectively escape the constrictions of serializability. Another roadblock, which has not been appreciated up to now, is that update operations on multimedia data will be increasingly common. So far, playout management has concerned previously existing presentations. However, there is no reason that the output of a database operation (query or otherwise) should not be a multimedia presentation constructed on the fly from existing presentations, perhaps with different presentation properties from those of their sources.

Buffer management. Continuous media presentations for many concurrent users require sophisticated buffer management techniques to deliver information on demand. When a multimedia object resides in the buffer, it should be shared among as many users as possible. However, it is difficult to schedule the buffering of such objects to maximize sharing and support user interactivity without violating the synchronization requirements of each presentation.

The traditional multimedia buffer replacement strategies do not perform well in an environment that supports sharing and interactivity. These strategies simply flush already presented multimedia data so the next multimedia object to be presented can be loaded. Traditional buffer replacement strategies, such as Least Recently Used (LRU), also do not work in an environment in which the access pattern history must be accounted for. Suppose we have an initially empty buffer that can hold 100 video frames. Suppose the user views frames 1–150 of a video. This viewing ends with the buffer containing frames 51–150. Now suppose a user wants to view frames 50–150 of the same video. The LRU strategy results in replacing frame 51 in the buffer with frame 50. After presenting frame 50, in order to view frame 51, frame 52 in the buffer is replaced by frame 51. For each successive frame viewed, there is a buffer fault.

Most current research concentrates on buffer management schemes for single-user, noninteractive presentations. There has been relatively little work discussing ways to manage user-based sharing and interactivity in a multimedia environment. For such sharing, buffers used for a user's presentation can be reused by others needing a similar presentation within some given time span. A technique called "least/most relevant for presentation" has been presented as an approach for interactive presentations. It recognizes that users may have set various bookmarks in the presentation to which they may want to jump, as well as the fact that the users are executing such commands as *fast-forward* or *play-backward*.

Research is just getting started on buffer management schemes for standard interactive multimedia. However, virtually no work has been done in database support for virtual reality environments, in which the amount of data and the choices for interaction are much greater than in standard multimedia environments.

Storage management. Storage of multimedia objects is not straightforward. Disk speeds have increased much more slowly than processor and primary memory speeds. The challenge is to serve multiple requests for multiple media streams so as to guarantee that the playout processes do not starve, while minimizing the buffer space needed and the time between an initial request for service and the time when the first bits of data become available [3]. Such techniques as data striping/interleaving, data compression, data contiguity, and storage hierarchies have been employed to reduce this bottleneck. Data striping/interleaving allocates space for a multimedia object across several parallel devices, whereas contiguity-based approaches try to store related multimedia objects contiguously on a single device. Also studied are storage hierarchies in which tertiary storage can be used for less frequently used or higher-resolution multimedia objects and faster devices for more frequently used or lower-resolution multimedia objects.

Recovery. Many advanced transaction models have generalized recovery methods. In a long, cooperating design environment, undoing complete transactions is quite wasteful, as a potentially large amount of work, some of it correct, might have to be undone. It makes much more sense to remove the effects of individual operations. To do this, however, the log must contain not only the history of the transaction but the individual operation-dependencies of the history.

Some advanced transaction models for long-running activities include compensating transactions for undoing the effect of an already committed transaction and contingency transactions, which provide an alternative to another transaction that could not be committed due to some failure condition. In a multitransaction that plays a multimedia presentation, a contingency transaction for showing a gif image might show a JPEG image.

In playout management, the notion of recovery extends to how to compensate for a presentation with unacceptable QoS. Besides mathematical complexi-

ties, compensation entails studies in human perception of multimedia that identify changes in a presentation imperceptible to the viewer.

Security. There has also been little work concerning multimedia-specific issues in database security, although multilevel security issues in hypermedia systems have been addressed [12]. Such a multilevel security model classifies documents into such levels as "Secret" and "Top Secret" and formulates various rules concerning the security levels of related documents. With an appropriate object-oriented decomposition of the universe of multimedia objects, it should be possible to construct a multilevel security model for such complex objects as a video using these techniques. However, much work has to be done to formalize the presentation of multimedia objects with various components either missing or transformed to hide various pieces of information. Selective editing is a far cry from not showing an unauthorized user certain field values of a record in a relational database system.

Commercial Systems for Multimedia Information Management

In the past, heated discussions among researchers in the multimedia computing and database communities concerned whether the then-current database systems could manage multimedia information [7]. On balance, people in multimedia computing were of the opinion that advances were needed in the database arena in order to manage this new type of data, whereas people in databases seemed to feel the newer database architectures were sufficient for the task.[2] Database architectures have surely changed from then to now, but there should be no argument that no existing database system contains all of the advanced options discussed in this article. Despite such limitations, there are today at least three commercial systems for visual information retrieval[3] and several commercial database systems[4] at various levels on the object-relational scale that manage multimedia information at an acceptable level. However, what is acceptable by today's standards will surely not be acceptable by tomorrow's.

For database systems to handle multimedia information efficiently in a production environment, some standardization has to occur. Relational systems are efficient because they have relatively few standard operations. The study of these operations by database researchers for many decades has resulted in numerous efficient implementations. Today, blades, cartridges, and extenders for multimedia information are designed in a completely ad-hoc manner. They work, but no one pays much attention to their efficiency. Operations on multimedia must become standardized and extensible. If the base operations become standardized, researchers can devote themselves to making them efficient; if extensible, complex operations can be defined in terms of simpler ones and still preserve efficiency. Hopefully, the efforts being devoted to MPEG-7 will address these concerns.

[2]Some researchers feel this is an example of the old adage that when all you have is a hammer, everything looks like a nail.

[3]Excalibur Technologies http://www.excalib.com; IBM http://www.ibm.com; Virage, Inc. http://www.virage.com

[4]CA-Jasmine http://www.cai.com/products/jasmine.htm; DB2 Universal Database http://www.ibm.com; Informix http://www.informix.com; ODB II http://www.datamation.com/PlugIn/inserts/FOSSI/ODB2/odb2main.html; Oracle http://www.oracle.com; Sybase http://www.sybase.com; and UniSQL http://www.unisql.com.

References

1. Chaudhuri, S., and Gravano, L. Optimizing queries over multimedia repositories. In *Proceedings of SIGMOD '96* (Montreal, Canada, June 1996). ACM Press, New York, 1996, pp. 91–102.
2. Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. Efficient and effective querying by image content. *J. Intell. Inf. Syst. 3*, 4 (1994), 231–262.
3. Gemmell, D., Vin, H., Kandlur, D., Rangan, P., and Rowe, L. Multimedia storage servers: A tutorial. *IEEE Comput. 28*, 5 (May 1995), 40–49.
4. Grosky, W. Multimedia information systems. *IEEE Multimedia 1*, 1 (Spring 1994), 12–24.
5. Grosky, W., Fotouhi, F., and Jiang, Z. Using metadata for the intelligent browsing of structured media objects. In *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Media*, A. Sheth and W. Klas, Eds. McGraw-Hill, New York, 1997, pp. 67–92.
6. Jagadish, H. Content-based indexing and retrieval. In *The Handbook of Multimedia Information Management*, W. Grosky, R. Jain, and R. Mehrotra, Eds. Prentice-Hall PTR, Upper Saddle River, N.J., 1997, pp. 69-93.
7. Jain, R. NSF Workshop on visual information management systems. *Sigmod Rec. 22*, 3 (Sept. 1993), 57–75.
8. Ramamritham, K., and Chrysanthis, P. *Executive Briefing: Advances in Concurrency Control and Transaction Processing*. IEEE Computer Society, Los Alamitos, Calif., 1997.
9. Stonebraker, M. *Object-Relational DBMSs: The Next Great Wave*. Morgan-Kaufmann, San Francisco, 1996.
10. Thimm, H., and Klas, W. δ-sets for optimized reactive adaptive playout management in distributed multimedia database systems. In *Proceedings of the 12th IEEE International Conference on Data Engineering* (New Orleans, Feb. 26–Mar. 1, 1996). IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 584–592.
11. Thimm, H., and Klas, W. Playout management in multimedia database systems. In *Design and Implementation of Multimedia Database Management Systems*, K. Nwosu, P. Berra, and B. Thuraisingham, Eds., Kluwer Academic Publishers, Boston, 1996, 318–376.
12. Thuraisingham, B. Multilevel security for information retrieval systems II. *Inf. Manage. 28*, 1 (Jan. 1995), 49–61.

William I. Grosky (grosky@cs.wayne.edu) is a professor in and chairman of the Computer Science Department of Wayne State University in Detroit. Beginning January 1998, he will also be Editor-in-Chief of IEEE's *Multimedia* magazine.